Linear Regression

Jose Luis Contreras Santos, Antonio Javier González Ferrer, Alejandro González Pérez

enero, 2017

Abstract

An attempt to use linear regression to predict wine quality.

Introduction

The aim of this section is to use linear regression to try and predict a wine's quality based on its chemical attributes. To do this, we will consider quality as a continuous variable ranging from 0 to 10. It is worth noting, however, that this variable is in fact a categorical one which can only take one of the 11 integer values comprised between 0 and 10. Therefore, as predictions will be continuous, many of them will most likely be slightly off due to this discrepancy. This should not pose a problem, however, as the usual error metrics, such as RMSE, can be obtained anyways.

Model fitting

After splitting the data in training and test sets, the training set was used to fit a linear regression model. At first, all variables were used as explanatory variables for the model.

```
# Train and test dataset, split 80% (data has been shuffled previously, no need to sam
split = nrow(df)*0.8
train = df[1:split,]
test = df[split:nrow(df),]

# Fit the model using all variables
model1 = lm(quality~fixed_acidity+volatile_acidity+citic_acid
            +residual_sugar+chlorides+free_sulfur_dioxide
            +total_sulfur_dioxide+density+pH+sulphates+alcohol, data=train)
summary(model1)
```

```
##
## Call:
## lm(formula = quality ~ fixed_acidity + volatile_acidity + citic_acid +
##     residual_sugar + chlorides + free_sulfur_dioxide + total_sulfur_dioxide +
##     density + pH + sulphates + alcohol, data = train)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8027 -0.4544 -0.0414  0.4593  2.7242
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.556e+01  1.399e+01   3.973 7.21e-05 ***
## fixed_acidity         6.376e-02  1.793e-02   3.556 0.000380 ***
## volatile_acidity     -1.304e+00  8.613e-02 -15.136  < 2e-16 ***
## citic_acid           -1.199e-01  8.904e-02  -1.347 0.178173
## residual_sugar        4.162e-02  5.878e-03   7.080 1.64e-12 ***
## chlorides            -4.269e-01  3.602e-01  -1.185 0.236015
## free_sulfur_dioxide   6.192e-03  8.400e-04   7.371 1.96e-13 ***
## total_sulfur_dioxide -2.547e-03  3.093e-04  -8.236 2.23e-16 ***
## density              -5.481e+01  1.427e+01  -3.842 0.000124 ***
## pH                    4.673e-01  1.028e-01   4.546 5.59e-06 ***
## sulphates             7.452e-01  8.423e-02   8.847  < 2e-16 ***
## alcohol               2.663e-01  1.957e-02  13.609  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7338 on 5184 degrees of freedom
## Multiple R-squared:  0.2918, Adjusted R-squared:  0.2903
## F-statistic: 194.1 on 11 and 5184 DF,  p-value: < 2.2e-16
```

However, as the summary above indicates, the p-values for citic acid and chlorides are not low enough to reject the null hypothesis. Thus, we can consider that the contribution of these to the variance of quality is not significantly greater than 0, and so they have been removed from the final model. The resulting linear regression model is described below.

```
# Citic_acid removed from the model
model1 = update(model1, ~.-citic_acid)
# Check if chlorides can also be removed
summary(model1)
```

```
##
## Call:
## lm(formula = quality ~ fixed_acidity + volatile_acidity + residual_sugar +
##     chlorides + free_sulfur_dioxide + total_sulfur_dioxide +
##     density + pH + sulphates + alcohol, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7891 -0.4580 -0.0373  0.4598  2.7323
##
```

```
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.554e+01  1.399e+01   3.971 7.26e-05 ***
## fixed_acidity         5.809e-02  1.743e-02   3.333 0.000866 ***
## volatile_acidity     -1.259e+00  7.940e-02 -15.852  < 2e-16 ***
## residual_sugar        4.136e-02  5.876e-03   7.038 2.20e-12 ***
## chlorides            -5.103e-01  3.549e-01  -1.438 0.150475
## free_sulfur_dioxide   6.232e-03  8.395e-04   7.424 1.33e-13 ***
## total_sulfur_dioxide -2.614e-03  3.053e-04  -8.563  < 2e-16 ***
## density              -5.479e+01  1.427e+01  -3.840 0.000124 ***
## pH                    4.770e-01  1.025e-01   4.652 3.38e-06 ***
## sulphates             7.409e-01  8.417e-02   8.801  < 2e-16 ***
## alcohol               2.642e-01  1.951e-02  13.544  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7339 on 5185 degrees of freedom
## Multiple R-squared:  0.2915, Adjusted R-squared:  0.2901
## F-statistic: 213.3 on 10 and 5185 DF,  p-value: < 2.2e-16
```

```r
# It can indeed be removed, do it
model1 = update(model1, ~.-chlorides)
summary(model1)
```

```
##
## Call:
## lm(formula = quality ~ fixed_acidity + volatile_acidity + residual_sugar +
##     free_sulfur_dioxide + total_sulfur_dioxide + density + pH +
##     sulphates + alcohol, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7877 -0.4560 -0.0367  0.4578  2.7385
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)           5.958e+01  1.370e+01   4.348 1.40e-05 ***
## fixed_acidity         6.160e-02  1.726e-02   3.569 0.000362 ***
## volatile_acidity     -1.275e+00  7.863e-02 -16.211  < 2e-16 ***
## residual_sugar        4.318e-02  5.738e-03   7.526 6.15e-14 ***
## free_sulfur_dioxide   6.185e-03  8.390e-04   7.372 1.94e-13 ***
## total_sulfur_dioxide -2.576e-03  3.042e-04  -8.469  < 2e-16 ***
## density              -5.899e+01  1.397e+01  -4.224 2.44e-05 ***
## pH                    5.068e-01  1.004e-01   5.046 4.67e-07 ***
## sulphates             7.161e-01  8.240e-02   8.690  < 2e-16 ***
```

```
## alcohol               2.637e-01  1.951e-02  13.517  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7339 on 5186 degrees of freedom
## Multiple R-squared:  0.2912, Adjusted R-squared:   0.29
## F-statistic: 236.8 on 9 and 5186 DF,  p-value: < 2.2e-16
```

According to the obtained model, the two variables with the highest influence on quality are volatile acidity and density, specially the latter. This can be misleading however, as it represents the variation in quality per unit of the input, and the scales of the input variables differ in their order of magnitude. Checking the distribution of *density*, for example, the difference between the maximum and minimun values is lower than 0.03, whereas $total_sulfur_dioxide$ has a range of over 400. In any case, some conclusions can be extracted from this output. For example, denser wines tend to have a higher perceived quality, and the same happens to those with higher alcohol contents.

The adjusted R-squared coefficient of the resulting model is rather low, with a value of only 0.29. Its associated p-value, however, is low enough for us to be confident that there exists a relationship between at least part of the input and the output variable, *quality*.

Assessing the model: assumptions

Let us further examine the quality of the linear regression model by checking if the assumptions are met. In particular, we are interested in testing if the residuals are independent, normal and have constant variance.

```
# A: Check the assumptions
# Test normality
jarque.bera.test(residuals(model1)) # Answer: No normality
```

```
##
##  Jarque Bera Test
##
## data:  residuals(model1)
## X-squared = 287.77, df = 2, p-value < 2.2e-16
```

```
# Equal variances
bptest(model1) # No constant variance
```

```
##
##  studentized Breusch-Pagan test
##
## data:  model1
## BP = 79.793, df = 9, p-value = 1.777e-13
```

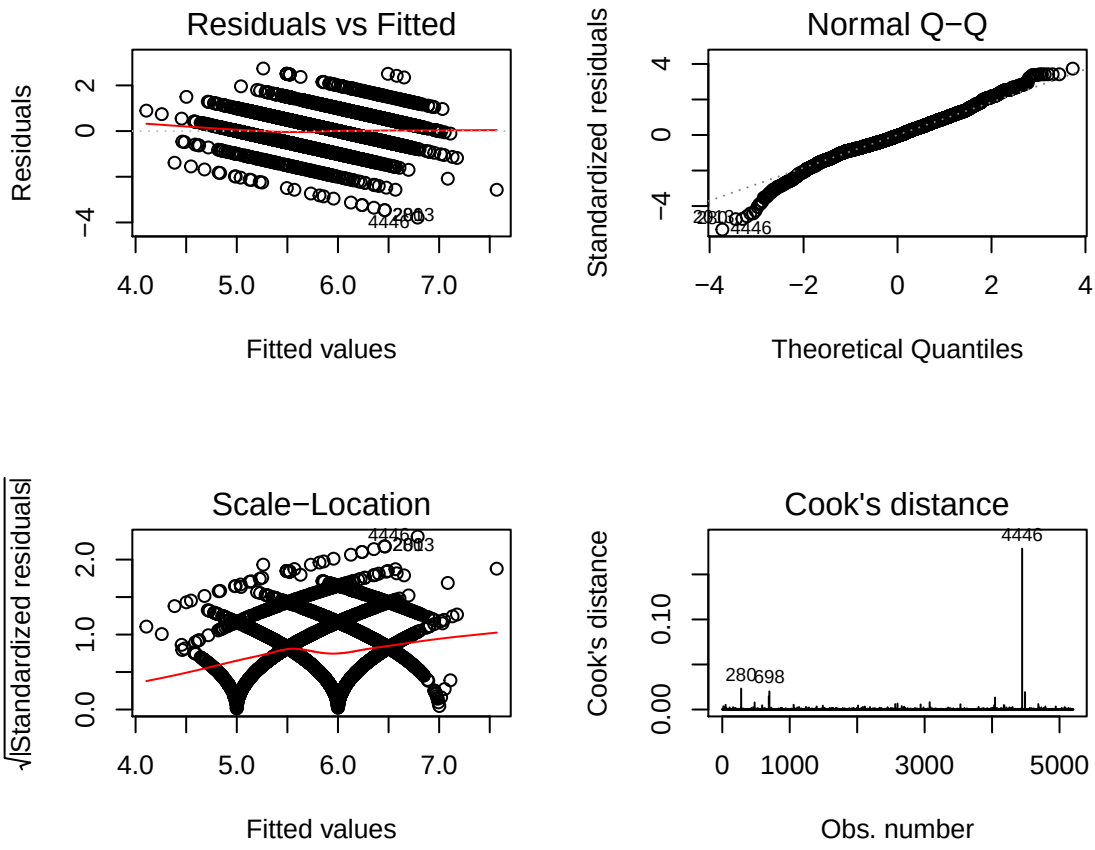


Figure 1: Residual plot

```
# Testing independence of the residuals
Box.test(residuals(model1)) # Independent
```

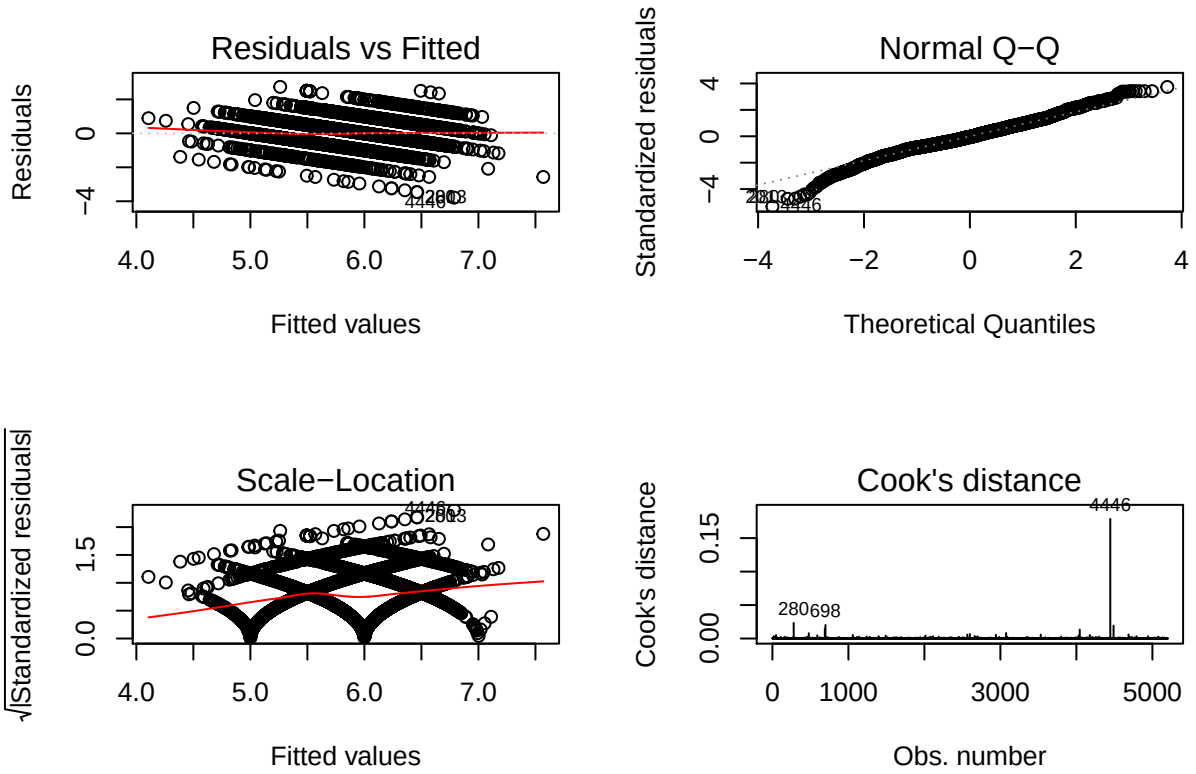
```
##
##  Box-Pierce test
##
## data:  residuals(model1)
## X-squared = 0.1241, df = 1, p-value = 0.7246
```

The conducted tests have different results. On the one hand, the Box-Pierce test returns a high p-value, sign that there is not enough evidence to reject the null hypothesis of independence. The results for the Jarque-Bera and Breusch-Pagan tests, on the other hand, are not so positive. According to their results, residuals are neither normal nor homoscedastic.

Although the residual plots are unusual due to the clustering around the integer values, they confirm the previous results. The Normal Q-Q plot is in line with the results of the Jarque-Bera test, showing departures from normality, specially in the first quartiles. From the Cook's distance plot, a significant outlier can be seen, corresponding to observation 4446.

Evaluation of results

```
# Residuals
par(mfrow=c(2,2))
plot(model1, which=c(1:4), ask=F)
```




```
# Un ojo a ese outlier gigante

# Conclusiones: en cuanto a hipotesis nuestro modelo es bastante mierda. Ademas parece
# para qualitys entre 4 y 6 mas o menos, los valores mas grandes van mal. Esto es norm
# muestra que tenemos es bastante mala (hacer histograma)

# 2. Assess the model. B: Evaluate performance
predicted <- predict(model1, test)

# Percentage of correct predictions aka accuracy (we are rounding the predicted variab
# Around 50% are correct - not bad
sum(test$quality == round(predicted))/nrow(test)
```

```
## [1] 0.5146154
```

```
# But the above is a simplification, we are treating quality as a continuous var, so l
# more conventional metrics, in this case rmse
rmse <- (sqrt(mean(test$quality - predicted)^2))
```

```
rmse
```

```
## [1] 0.006010179
```

```
# Pseudo rmse (after discretizing the output var by rounding)
# Aumenta bastante, pero ahora - yo diria - es una cifra mas indicativa
rmse.discret <- (sqrt(mean((test$quality - round(predicted))^2)))
rmse.discret
```

```
## [1] 0.8133501
```

```
# Conclusiones: No funciona mal en cuanto a resultados, pero no nos fiamos, entre que
# datos son malillos (en cuanto a distribucion) y que no hemos usado la tecnica adecua
# En cualquier caso, posiblemente seria mejor usar Logistic Regression (o algun metodo
# para predecir la quality, que realmente no es continua.
```